AIT - WordToTFS User Guide

V3.1.1405.1301

Contents

System Requirements 2
Quick Start: Installation 2
Quick Start: Overview 2
Quick Start: Add a Template Location 4
Quick Start: Connect, Create, Publish 5
Product Usage 6
Connect to a Team Project 6
Choose a Template 6
Manage Available Templates 7
Insert Work Items 8
Support for any kind of work item 9
Work Item Overview 9
Publish work items to TFS 9
Edit default values for the current template 11
Delete work item ids from an existing document 12
Import Existing Work Items from TFS 12
Insert Area and Iteration Path Values 14
Refresh all or selected work items 15
Using Headers 15
Provide Feedback 16
Test Management Reports 17
Structure of generated report 19
Managing Templates 21
Add a Template Location 21
Customizing the template configuration: w2t file 23
Creating a new work item mapping 24
Defining Headers 31
Customizing the template layout: .xml files 33
Configuring the test reporting extension 33
General test configuration 33
Test specification report configuration 34

Test result report configuration 34
Templates 35
Invoke methods from within the template 36
Object used to resolve property value 36
Object model extensions 37
Hyperlink configuration 37
Linked templates 38
Enhanced Document Structure

FAQ 39

System Requirements

AIT WordToTfs supports Word 2007 and Word 2010. Make sure you have the following software installed before trying to attempt to install the WordToTfs add-on:

.NET Framework 4.0 + Windows Installer 3.1 + Team Explorer 2012

For more information please go to: http://wordtotfs.aitgmbh.de

Quick Start: Installation

You can choose to directly install WordToTFS via Click Once from our webserver, or to install it through the offline setup. Just follow the installation wizard.

WordToTFS will receive updates automatically as long as you have an active Internet connection.

Just refer to http://wordtotfs.aitgmbh.de for the latest news and updates.

Quick Start: Overview

(1) Connect to a team project in TFS, if connected, shows server and project connected to, click again to disconnect

(2) Publish the changes from the document into the connect team project in TFS

(3) Opens the task pane to import existing work items from TFS into the document

(4) Dropdown to select a mapping template

(5) Opens template manager to manage template locations

(6) Inserts a blank work item table

(7) Inserts a work item table with default values

(8) Opens the task pane to edit default values for new work items (7)

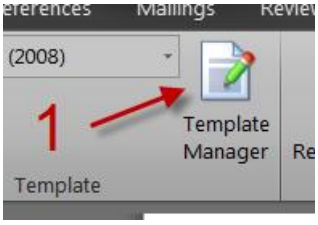
(9) Deletes all Ids from published work items to re-publish word file to another server

(10) Opens the Area and Iteration task pane to insert Area and Iteration Paths into the document

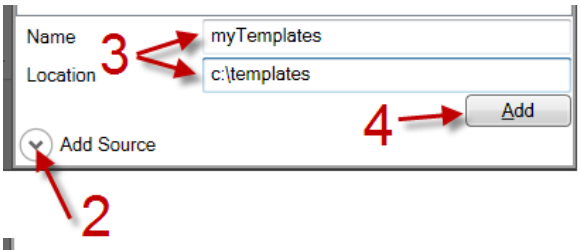
(11) Opens licensing dialog

(12) ***Help*** opens help file, ***About*** shows Version and ***Feedback*** opens a web form to give feedback to AIT

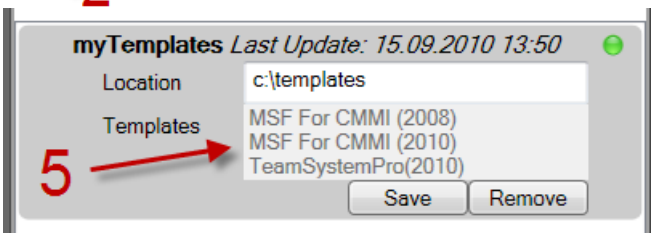
Quick Start: Add a Template Location



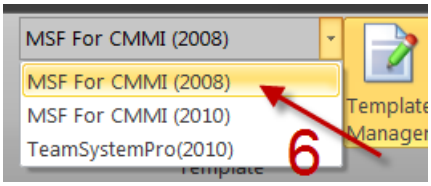
(1) Click the ***Template Manager*** button



(2) Click ***Add Source*** in the lower
(3) Enter name and path of the location
(4) Click ***Add***

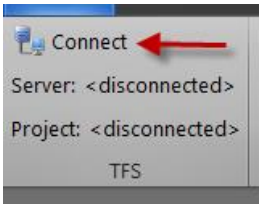


(5) Click on the List box entry, check all templates are shown correct

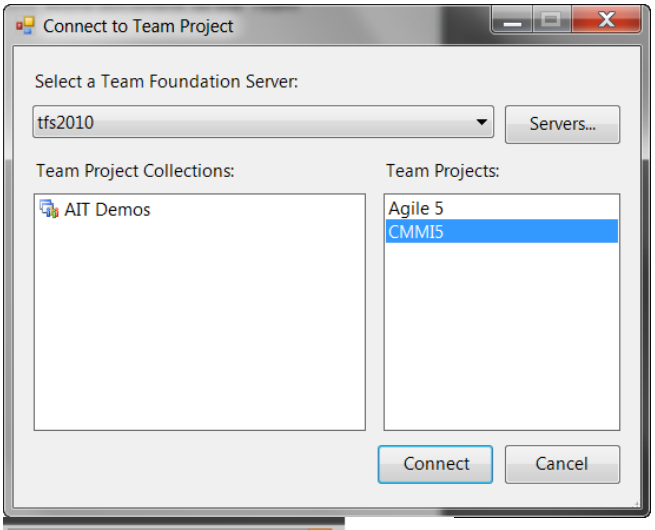


(6) Select the template to use for the current document from the Template combo box

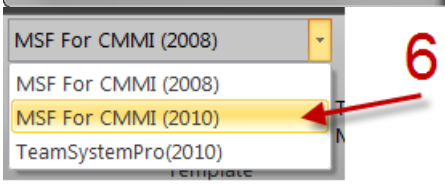
Quick Start: Connect, Create, Publish



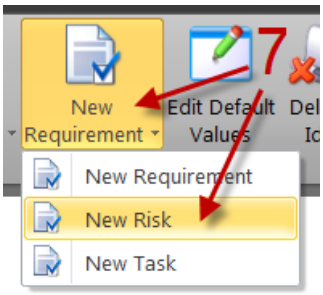
(1) Click ***Connect***



(2) Choose server
(3) Choose Team Project Collection
(4) Choose Team Project
(5) Click ***Connect***



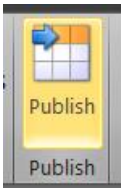
(6) Select Template



(7) Add some work items

Risk	Peter Projectmanager	
The risk of rain today	1	3
This is a high risk		

(8) Fill in the work item table with values according to your rule set



(9) Click ***Publish***

Product Usage

Connect to a Team Project

To connect WordToTFS 2010 with a Microsoft Team Foundation Server Team Project, press the ***Connect*** button in the WordToTFS Ribbon. A new dialog will open as shown in figure 1. Choose a Team Foundation Server, a Collection and a Team Project where the work items will be stored. Press the ***Connect*** button at the bottom of the dialog to connect the current Word document to the Team Foundation Server.

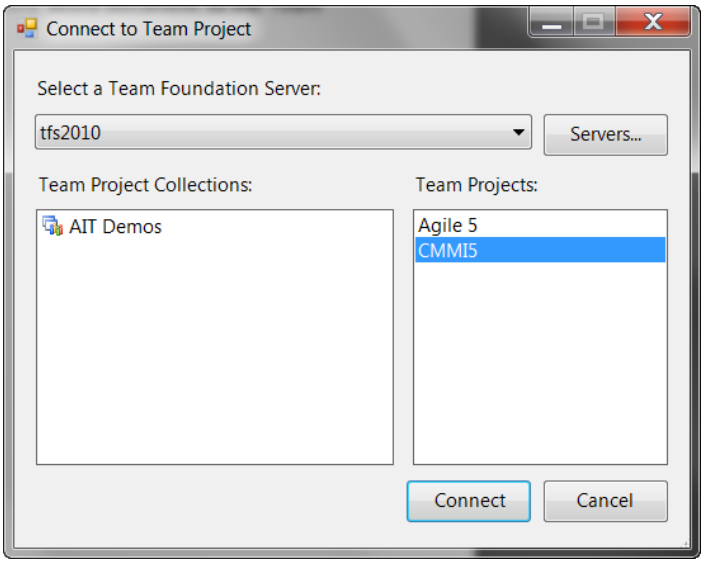
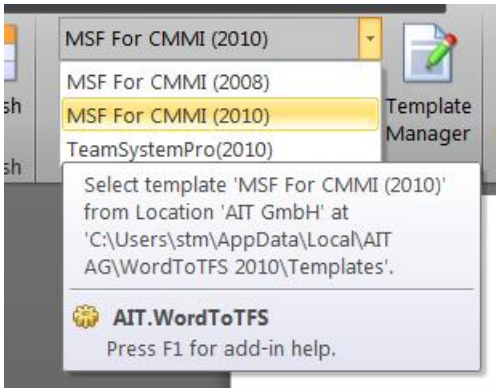


Figure 1 - Connect the Word document to a Team Foundation Server

(1) Press the ***Connect*** button in the ribbon.
(2) Select a Team Foundation Server from the drop down box or add a new Team Foundation Server by pressing the ***Servers…*** button.
(3) Select a Collection from the list of available Collections in the lower left part of the window.
(4) Select a Team Project from the list of available Team Projects in the lower right part of the window.
(5) Press ***Connect***

Choose a Template


MSF For CMMI (2010)
MSF For CMMI (2008)
MSF For CMMI (2010)
TeamSystemPro(2010)
Template Manager
Select template 'MSF For CMMI (2010)' from Location 'AIT GmbH' at 'C:\Users\stm\AppData\Local\AIT AG\WordToTFS 2010\Templates'.
AIT.WordToTFS
Press F1 for add-in help.

To select an appropriate template just select the right one from the dropdown in the templates section.

Note that the selected template has to match the Team Project type which WordToTFS is connected to. If the selected template contains work items or work item fields that are not supported by the Team Project you will be shown an addition dialog each time you change to this template or connect to another server.

Figure 2 - Template dropdown and Tooltip

Each template owns a tooltip which tells you about its source location.

After selecting the template, the defined fields in the template will be checked against the server you are connected to. In case of mismatching, Word to TFS will tell you about:

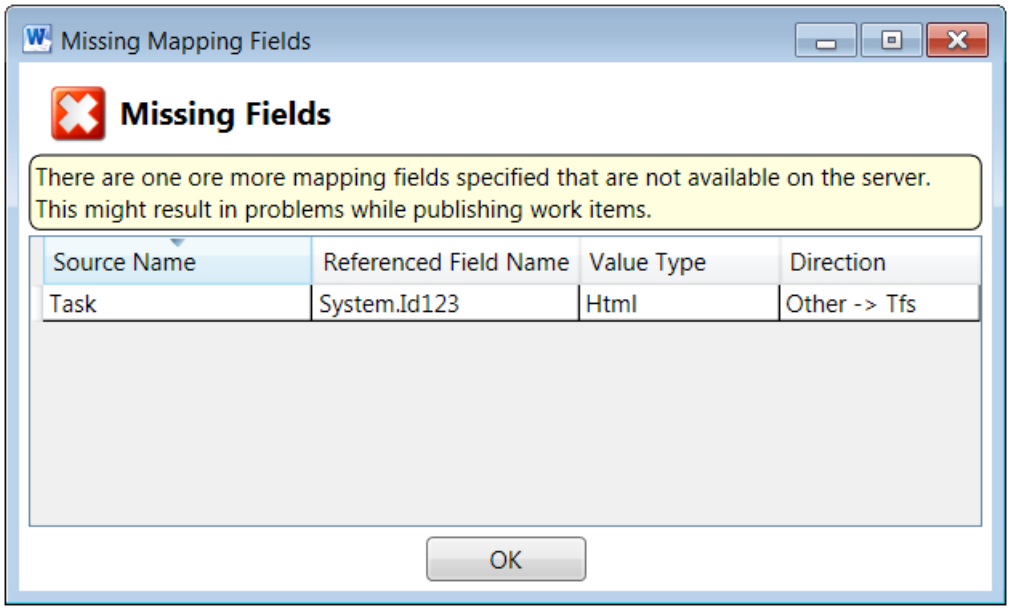


Figure 3 - Missing Mapping Fields Dialog

Even if there are missing fields, you should be able to synchronize with the TFS server. The ability of successfully synchronize depends on the type of missing field. If it is an optional field, the field value just isn´t saved on the TFS (but in the word file, of course) or in the other way round, it's not resynced to the work item table in Word. However, if it is a required field and the direction is word to TFS, you are not able to successfully sync to the TFS server.

Manage Available Templates

You can see the template manager button right next to the template selection dropdown. The template manager provides you independent use of different templates from different locations such as from your network, your local hard drive or even from the internet via http. After opening it, you can see your default template location provided by AIT. Select the list entry, to see further details.

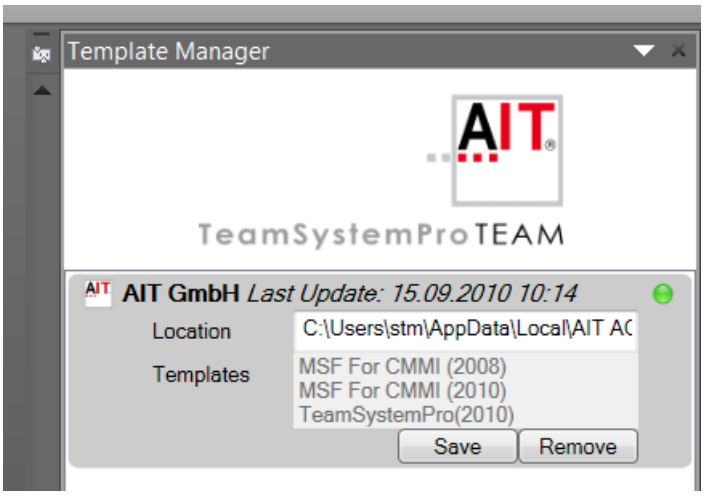


Figure 4 - The template manager and the default templates

To add a new location, click on the add Source expander on the lower side of the panel, enter your template location, and click add. WordtoTFS immediately starts to scan the folder for suitable template files, and add them after a quick check to the templates list. As already mentioned the path can be a UNC, local or even a web path. For providing web location ask your administrator or read the admin guide.

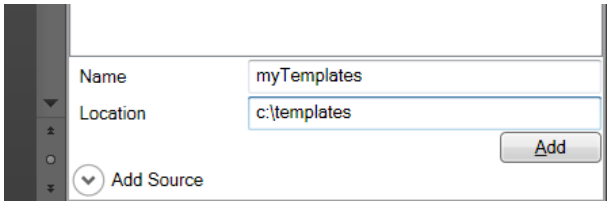


Figure 5 - Add a new template location

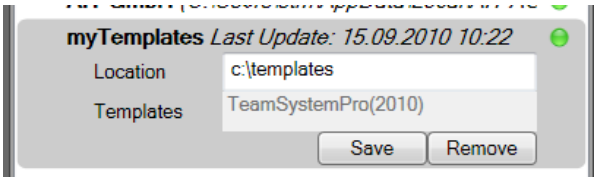


Figure 6 - The new added templates

Now you are able to select the new template from the templates combo box. (Figure 2)

Insert Work Items

Depending on the selected Template different Work Item Types are available for insertion into the document.

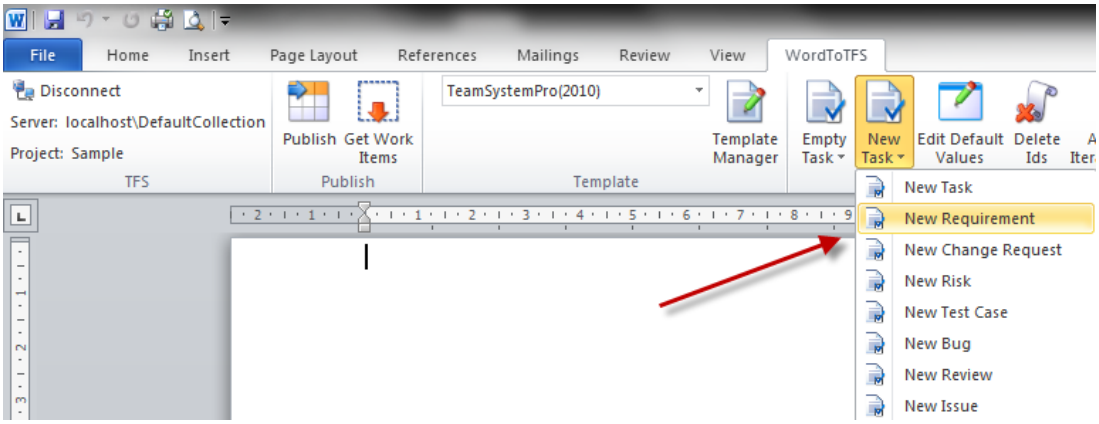


Figure 7 - Choosing from a list of work item types

A new Work Item is inserted as table into the document.

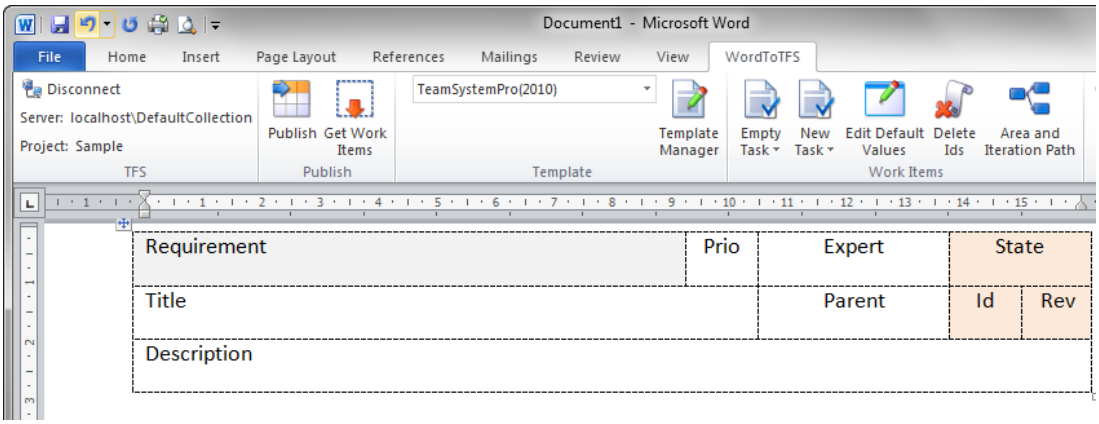


Figure 8 - The inserted work item table according to work item type and selected template

The table contains hints what should be inserted in each cell. The default values can be modified using the *Edit Default Values* task pane.

Support for any kind of work item

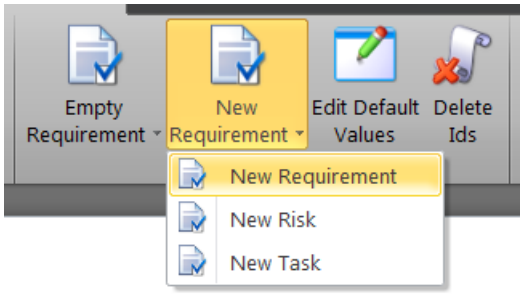
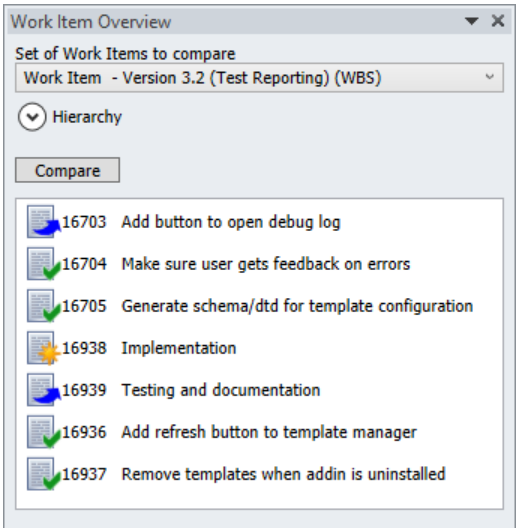


Figure 9 - Adding a new work item table

WordToTFS 2010 supports any kind of work items. The customizable *.w2t configuration files (see the WordToTFS 2010 - Admin Guide) is capable of describing any kind of work items, those predefined by Microsoft Team Foundation Server as well as custom created work items. In addition a single template may contain any number of different work item types. To select a specific work item type click on the ***New*** button to open a selection menu of all available work items for the selected template as shown in figure 3 and figure 4. Selecting one of the available work items will add a new table as defined in the work item configuration into the document and fills the table with default values. The ***Empty*** button provides the same functionality like the ***New*** button does, except the new created table will not contain any default values.

Work Item Overview

The work item overview allows you to compare the work item in your word document to the work item on your server. You can choose to either query all work items found in your word document, or you can pick one of your saved queries. Selecting a query will only compare work items found in both, the query and your document. It will also tell you, if the query returned work items which are not yet imported into your document.


Work Item Overview
Set of Work Items to compare
Work Item - Version 3.2 (Test Reporting) (WBS)
Hierarchy
Compare
16703 Add button to open debug log
16704 Make sure user gets feedback on errors
16705 Generate schema/dtd for template configuration
16938 Implementation
16939 Testing and documentation
16936 Add refresh button to template manager
16937 Remove templates when addin is uninstalled

Publish work items to TFS

To publish work items from the Word document to the connected Team Project press the publish button in the WordToTFS ribbon. (This button is only enabled if you are connected to a TFS!) Publishing will create new work items in the Team Project for each work item in the Word document that does not have an id yet. Any work items from the Word document that already have an id will be updated. You can see the state of the publishing process on the right side at a task pane, which opens automatically once you have started the publishing process.

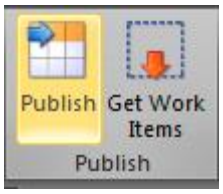


Figure 10 - Publish your changes to TFS

Figure 11 - The publish result

After the publish process has completed, all work items that are successfully saved to TFS are shown as list in the Publish task pane.

In case of any errors during the Publish, a red cross will indicate the error. Click the error entry in the Publish task pane to see details about

the issue. To navigate to the work item table that caused the error, use the magnifier icon at the bottom right corner of the error entry.

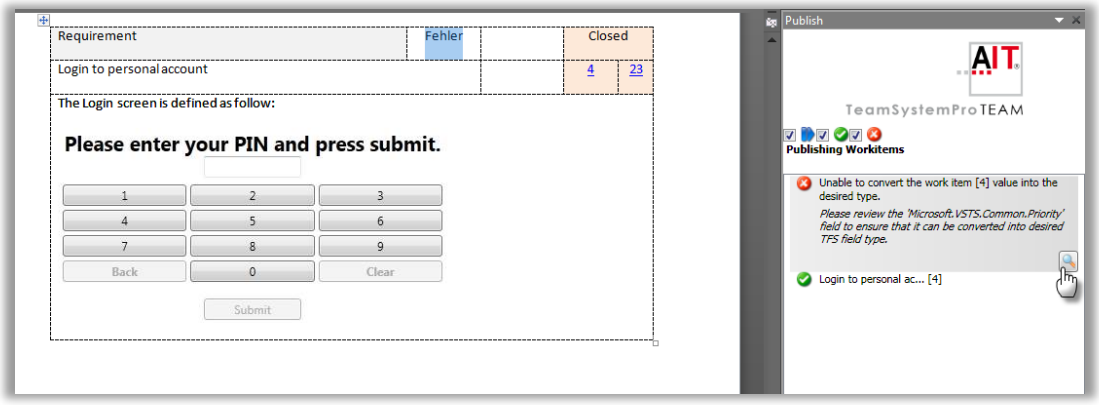


Figure 12 - Errors during Publish are listed inside the Publish task pane

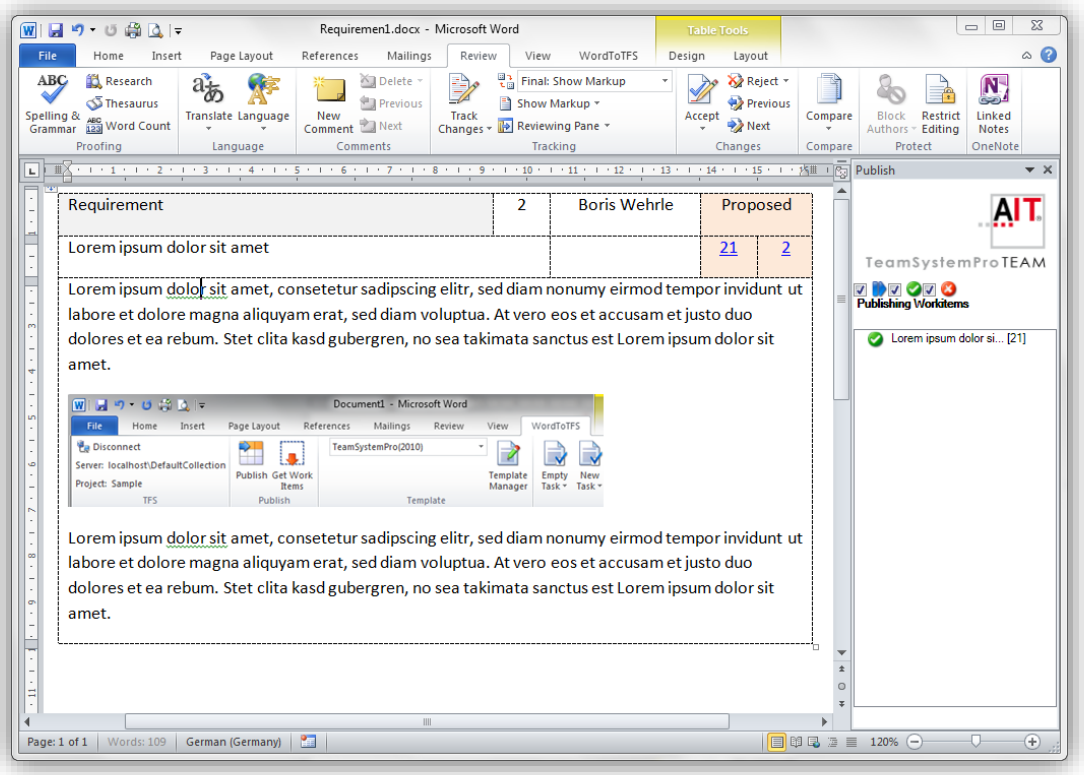

Requiremen1.docx - Microsoft Word
Table Tools
Requirement
2
Boris Wehrle
Proposed
Lorem ipsum dolor sit amet
21
2
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
Document1 - Microsoft Word
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
Publish
TeamSystemProTEAM
Publishing Workitems
Lorem ipsum dolor si... [21]

Figure 13 - Sample Requirement work item table

During publish the state, revision and work item id is inserted or updated inside the Word document.

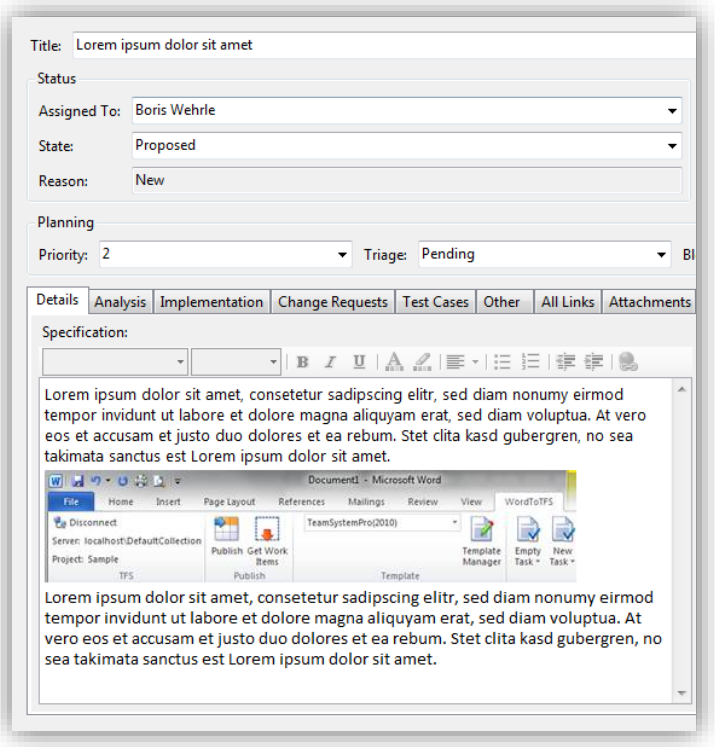
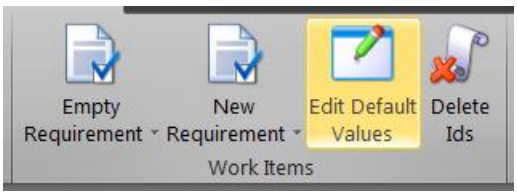

Title: Lorem ipsum dolor sit amet
Status
Assigned To: Boris Wehrle
State: Proposed
Reason: New
Planning
Priority: 2
Triage: Pending
Details
Analysis
Implementation
Change Requests
Test Cases
Other
All Links
Attachments
Specification:
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Figure 14 - The published work item in TFS

Edit default values for the current template



To edit the default values of the current template, press the button ***Edit Default Values***. At the right border of the window a new task pane will appear, containing all work items of the selected templates and all their fields.

Figure 15 - Edit default values

After editing the default values the window may be closed. Inserting a new work item using the ***New*** button will create a new table in the word document using the previously specified default values.

All default values can be reset by pressing the ***Reset Values*** button at the bottom of the default values window.

(1) Click the ***Edit Default Values*** button
(2) Edit the default values in the ***Default Values*** task pane at the right side of the screen
(3) Close the ***Default Values*** task pane

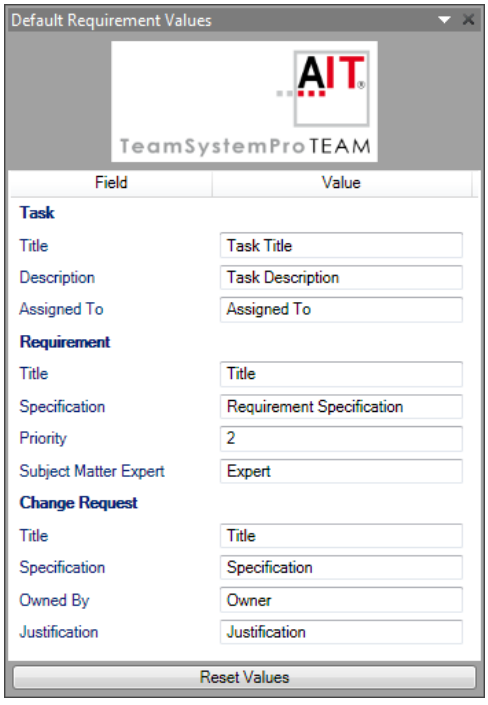


Figure 16 - A list of all editable fields and their default values

Delete work item ids from an existing document

By pressing the ***Delete Ids*** button as shown in figure 11, all fields of all tables in the document that points to a TFS work item ***System.Id*** field are cleared. This allows the synchronization of one document on different Team Foundation Servers.

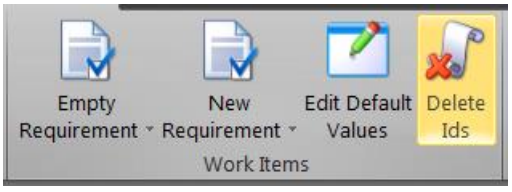


Figure 17 - Pressing the 'Delete Ids' button will delete all work item ids from the document

(1) Press the ***Delete Ids*** button
(2) Confirm the deletion of all ids

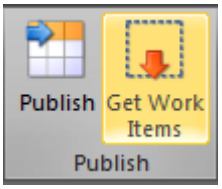


Figure 19 Get Work Items

Import Existing Work Items from TFS

You can import work items from TFS into your Word document by clicking ***Get Work Items***. This opens the ***Get Work Items*** task pane which gives you several options for getting work items from TFS.

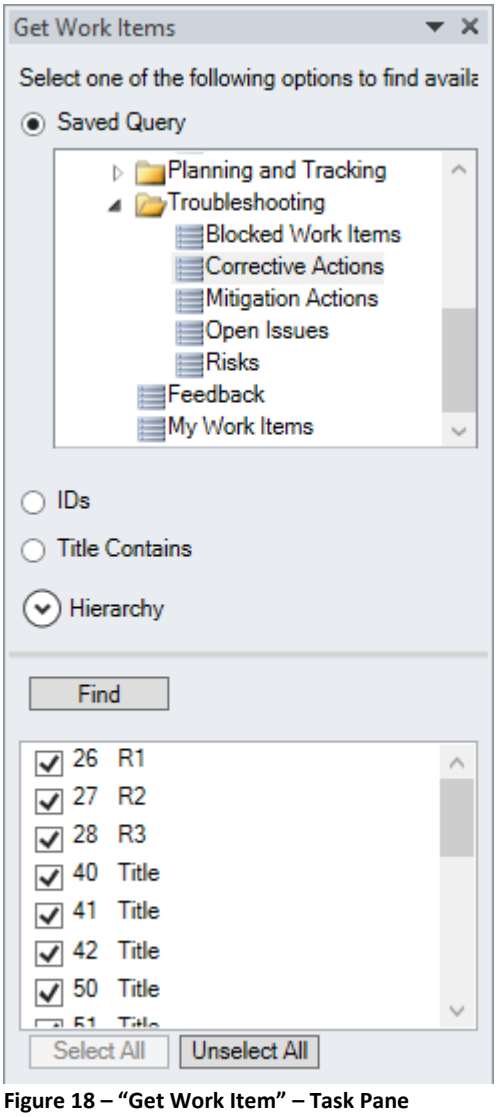


Figure 18 – "Get Work Item" – Task Pane

You can query work items by Work Item Query, IDs or title. You can also include linked work items based on work item links from the ***Hierarchy*** pane.

Click ***Find*** to get the work items matching your criteria. To insert the work items found, select them from the result list and click ***Import***. The work item tables will be inserted at the current cursor position within the Word document. In case work item tables already exist in the document, they will be updated (overwritten!) inside the document with the latest values from TFS.

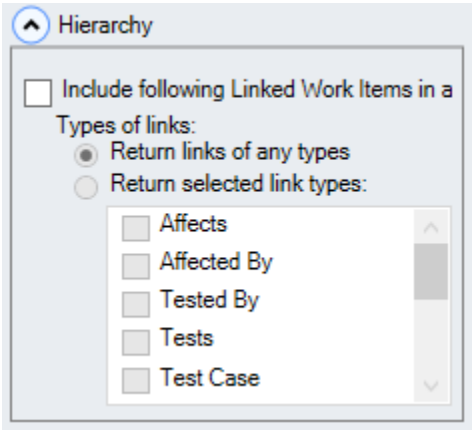


Figure 20 - Hierarchy Pane

Create Hyperlink creates hyperlinks to the selected work items at the cursor position.

After the import, the complete work item table is inserted into the document:

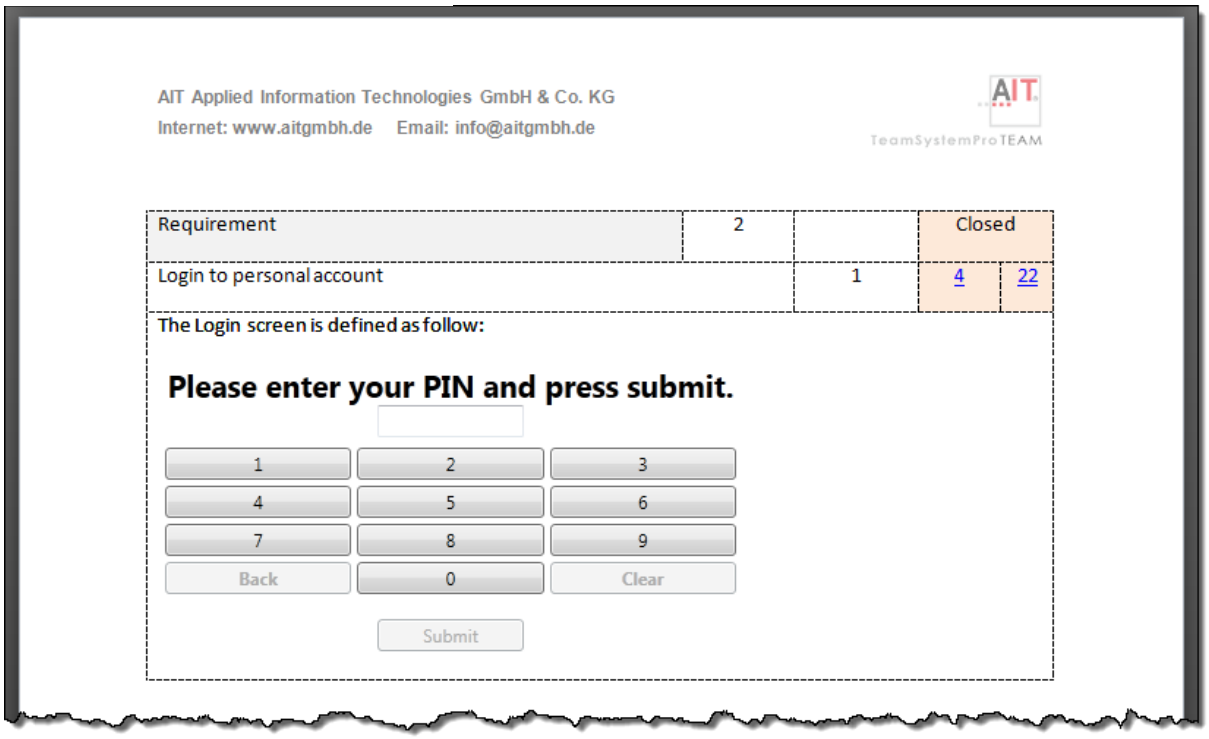


Figure 21 - A sample work item table in Word

The work item might look like this in TFS:

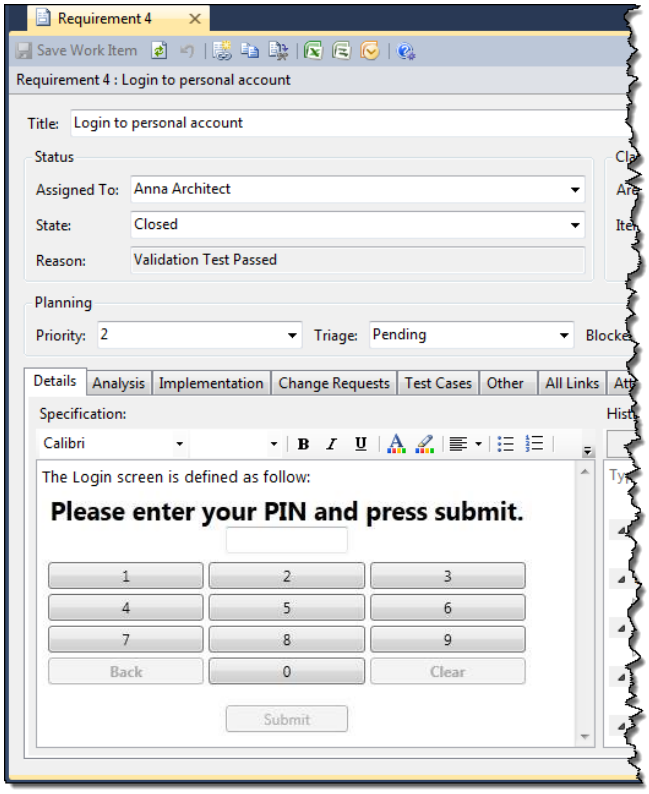
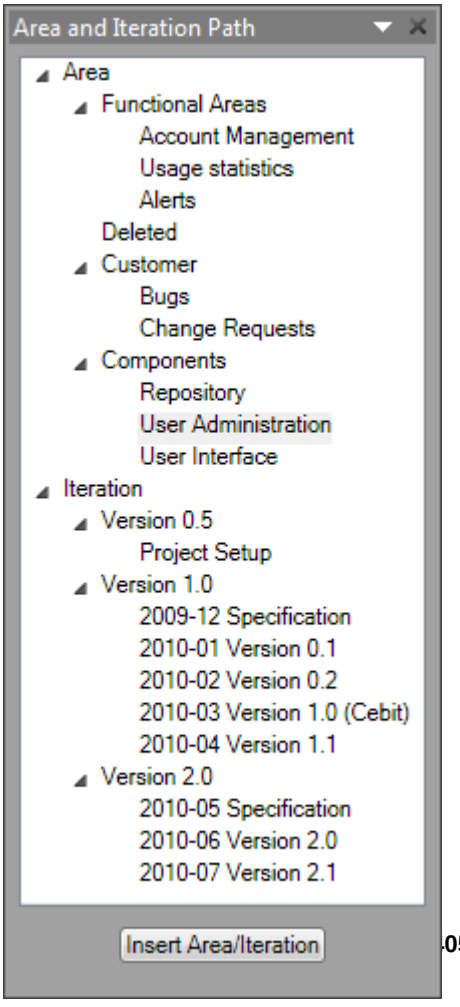


Figure 22 - Same sample in TFS

Insert Area and Iteration Path Values



You can insert Area and Iteration Path values within the document. This way, you will be able to set for instance the planned version or the feature area of your work items from within Word.

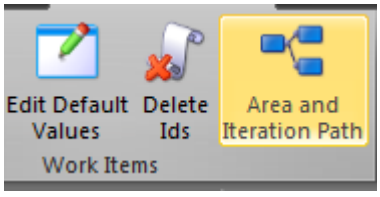


Figure 23 - View Area and Iteration task pane

Simply click ***Area and Iteration Path*** from the WordToTFS ribbon tab. This will open the Area and Iteration Path task pane shown on the left.

The Area and Iteration tree from the connected team project will be shown. Select an Area or Iteration Path node and click ***Insert Area/Iteration*** to add it at the current text cursor position into the document. This could be inside a work item table:

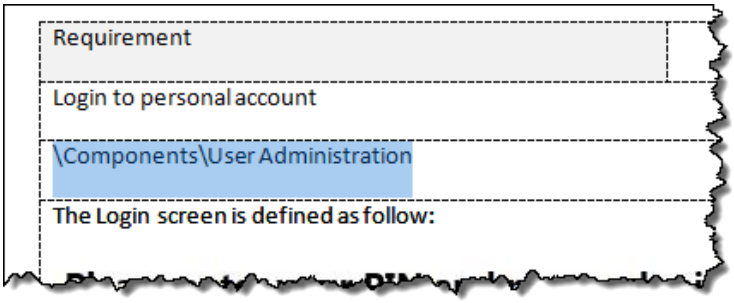


Figure 24 - The Area and Iteration task pane Figure 25 - Inserted Area Path node inside a work item table

Refresh all or selected work items

Figure 26 - Refresh all

In order to update all the work items inside the document with the latest contents from TFS, use the Refresh option from the WordToTFS ribbon tab.

You can also refresh individual work items by selecting the tables in Word and execute “Refresh selected work items.” from the drop-down Refresh menu.

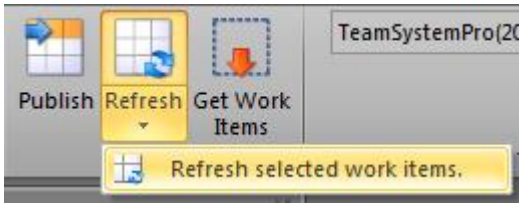


Figure 27 - Refresh selected only

Using Headers

Headers are document elements like work items that can be inserted anywhere outside of a table. Headers are used to set field values for multiple work items at once. Whether a work item has its field values set by a header depends on the position of the work item. Only work items in the header range are affected by its configuration and values.

The range of a header starts with the header itself and ends with a header of the same or a lower hierarchy level. Headers can be used to set fields for all work items within its range. This is especially useful if the fields of the header are not also mapped in the affected work items as it helps keeping the document more compact. You can for example define a header that only sets the area path for all work items and don’t have to add the area path field to other work item mappings. The button to insert headers is only visible if there are headers configured in the selected template.

Example:

With a correctly configured header, publishing the change request will automatically set the Area defined in the header for this work item. This helps keeping the document clean and easy to maintain, and is especially helpful when adding many items.

Area Header	This value applies to all affected work items
Area	\User Interface

Change Request	2	Developer	Proposed	
Add breadcrumbs to user navigation		12381 - Options	382	12
Tree view hierarchy is very deep and the tree view itself consumes too much space, better to replace it with breadcrumbs				

Provide Feedback

Figure 28 - Please Provide Feedback!

You need further information, or help? Or you want to use WordToTFS at your project but you need some special functionality at the product? Or do you need a specialized template for your process template? Or just want to leave us your ***Feedback***?

Contact us!

Test Management Reports

Figure 29: Test reporting ribbon

WordToTfs supports the generation of test specification reports and test result reports based on the information stored in TFS. The features are accessible via the buttons in the Test Management Ribbon group. Use the buttons shown in Figure 29 to show or hide one of the Test Management task panes

Test specification report

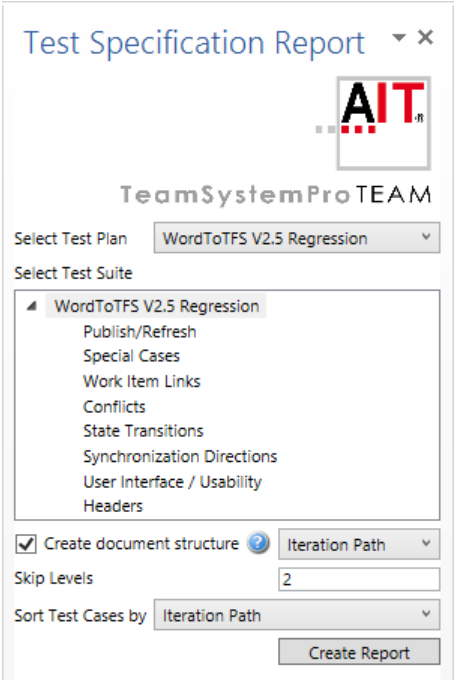


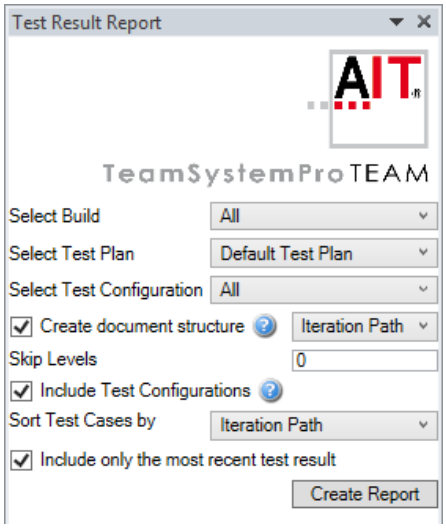
Figure 30: Test Specification Report task pane

The test specification reports enables the creation of reports that lists test cases and related information. Once the information is received from TFS and inserted into word it is disconnected from TFS which means any changes you do to the generated word content cannot be published to TFS again. Also the refresh option is not available for generated reports. The task pane shown in Figure 30 allows to configure how the report is generated. The following table explains the different options:

Term	Explanation
Select Test Plan	Allows the user to select the test plan that contains the test cases that should be used in the report.
Select Test Suite	The test suite hierarchy of the selected test plan is shown and the user has the option to restrict the test cases to use in the report.
Create document structure	This option enables the creation of word document structure.
Document structure	If the report should be structured, it is possible to choose one of these possibilities: • Iteration Path • Area Path • Test Plan Hierarchy
Skip Levels	If the report should be structured and the structure is either Iteration Path or Area Path, this number determines how many levels should be skipped in generated structure. See also chapter Structure of generated report.
Sort Test Cases by	Defines the criteria the test cases are sorted when added to the report. Available sort types are: • Iteration Path

	• Area Path • Work Item Id
Create Report	Clicking this button will start the generation of the report. The generation is started at the current cursor position.

Test result report



The test result report enables the creation of reports that lists the result of different test runs. The following table lists the options available in the test result report task pane:

Figure 31: Test Result Report task pane

Term	Explanation
Select Build	Allows to select a specific build to generate the test result report for
Select Test Plan	Allows the user to select the test plan that contains the test results that should be used in the report.
Select Test Configuration	Allows to restrict the test results used in the report to those being generated with a specific test configuration
Create document structure	This option enables the creation of word document structure.
Document structure	If the report should be structured, it is possible to choose one of these possibilities: • Iteration Path • Area Path • Test Plan Hierarchy
Skip Levels	If the report should be structured and the structure is either Iteration Path or Area Path, this number determines how many levels should be skipped in generated structure. See also chapter Structure of generated report.
Include Test Configurations	This option defines whether the test configurations should be included in the report.
Sort Test Cases by	Defines the criteria the test cases are sorted when added to the report. Available sort types are: • Iteration Path • Area Path • Work Item Id
Include only the most recent test result	If checked, for each test case only the most recent test result (DateCompleted) is reported.
Create Report	Clicking this button will start the generation of the report. The generation is started at the current cursor position.

Structure of generated report

The following section details how the word document structure is generated based on the users selection in the test reporting task panes.

Let's suppose the test cases are defined in following iterations:

- **Base** – 2 test cases (BaseTC1 and TaseTC2)
- **Base/Extended** – 0 test cases
- **Base/Extended/Iteration 1** – 3 test cases (Iteration1TC1, Iteration1TC2 and Iteration1TC3)
- **Base/Extended/Iteration 2** – 1 test case (Iteration2TC1)
- **Base/Extended/Iteration 3** – 0 test cases

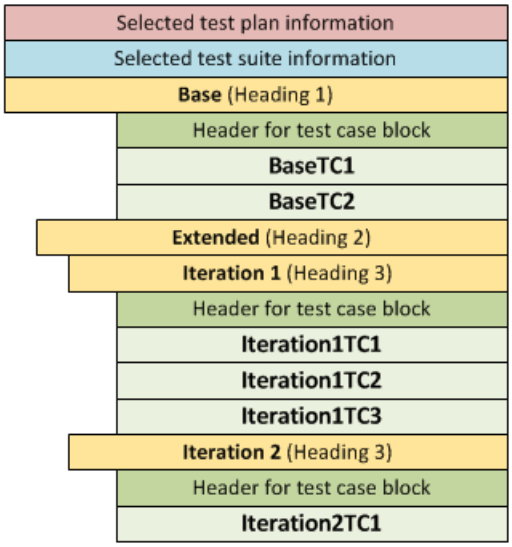


Figure 32: Skip Level 0

If the Skip Levels is set to 0, the structure of the generated report will be as shown in Figure 32.

Notes

- In generated report are used all test cases defined in selected test suite and all its child test suites.
- If the iteration contains no test case, header for test case block is not generated.
- If the iteration contains no test case and its child iteration too, iteration is not used in generated report

If the Skip Levels is set to 2, the structure of the generated report will be as shown in Figure 33.

Notes

- If the block is not shown, its test cases are not shown too.

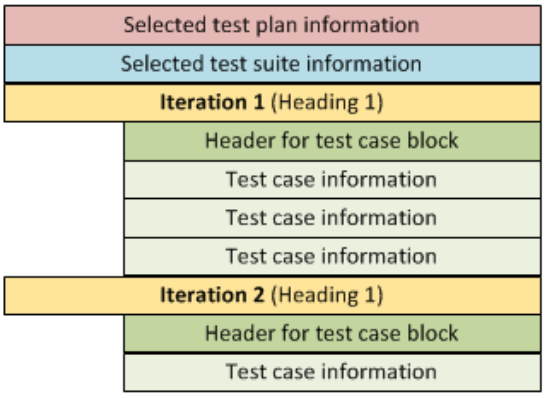


Figure 33: Skip Level 2

A structure based on the area path could be generated in a way equal to a structure based on iteration path.

Let's suppose the test cases are defined in following iterations:

- **Backend** – 0 test cases
- **Backend/Database** – 2 test cases (DatabaseTC1 and DatabaseTC2)
- **Backend/Services** – 3 test cases (ServicesTC1, ServicesTC2 and ServicesTC3)

If the Skip Levels is set to 0, the structure of the generated report will be as follows:

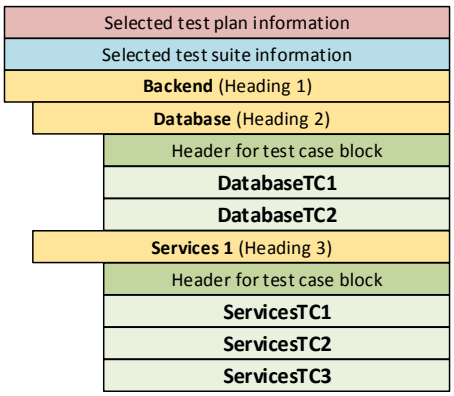


Figure 34: Test specification report structured by area and skip levels is set to 0

If the Skip Levels is set to 2, the structure of the generated report will be as follows:

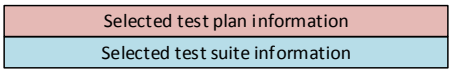


Figure 35: Test specification report structured by area and skip levels is set to 2

Note

- If no area exists for the filter, no test case is generated.

The last criteria for document generating a document structure is the test plan hierarchy. Supposed the test suites are defined in followed structure we would get a document structure as shown in Figure 36

- **Common** – 1 test case (CommonTC1)
  - **Publish** – 2 test cases (PublishTC1 and DatabaseTC2)
    - **Internet** – 1 test case (InternetTC1)
    - **Intranet** – 1 test case (IntranetTC1)
  - **Conflicts** – 2 test cases (ConflictsTC1)

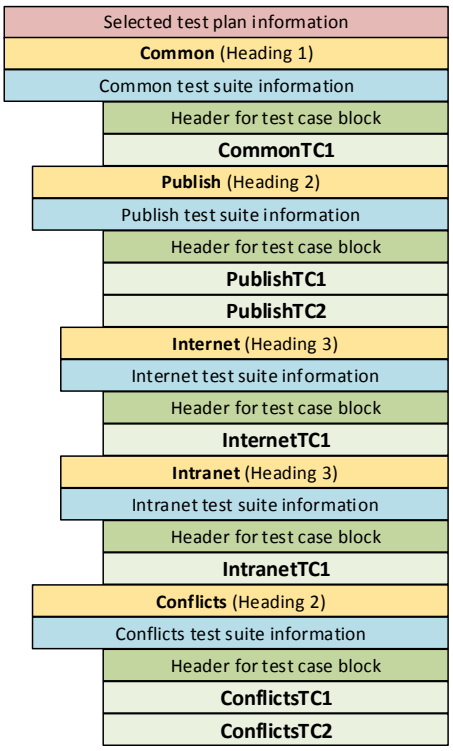


Figure 36: Test specification report structured by test plan hierarchy

Note

- *For every test suite block are used only test cases defined for according test suite.*

A final scenario is the unstructured report where no document structure is generated

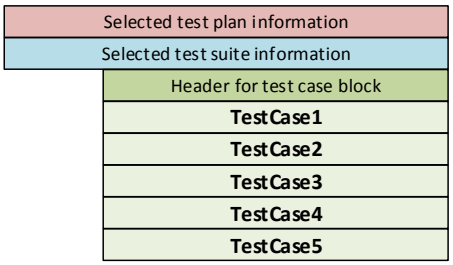


Figure 37: Test specification report unstructured

For the test result report the same mechanism is used but there are some additional restrictions:

1. The test configurations being used are inserted on the beginning of report.
2. Only test cases with at least one test result are included in the report
3. Under each test case are displayed its test results.

Managing Templates

WordToTFS comes with some default templates, which are shown under the “AIT GmbH” location in the templates manager. The template manager allows you to easily work with different configurations from different physical locations. The templates found at the given locations are cached, so you can use them even if the original source is not available.

Add a Template Location

When you add a template, you need to provide a directory as well as a name used to identify this template in the template manager and the template dropdown in the Word ribbon.

Local and UNC location

A typical location folder contains 3 different types of files. (See later in this document how to customize them) First of them the “w2t” file, one per template. The w2t file contains the mapping between the TFS server fields and the Word table cells. We deliver WordToTFS with the following templates per default: MSF for CMMI (2010), MSF for CMMI (2008) and TeamSystemPro(2010). The second type is the word-table files as .xml. They contain the design and look of each of the work item types, such as task, requirement, bug etc. The last one is a location avatar as ico file. If you want to show an icon in the template manager you have to add it to the location folder and rename it to “favicon.ico”.

Add a location at the template manager:

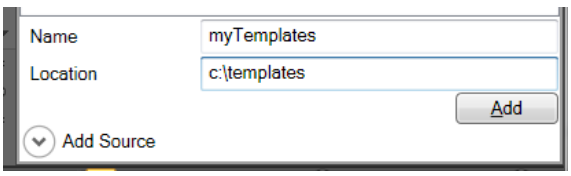


Figure 38 - Local directory

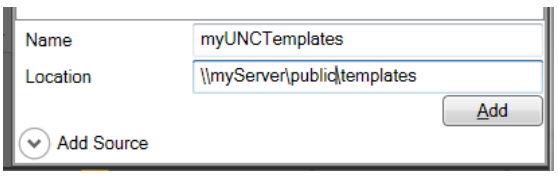
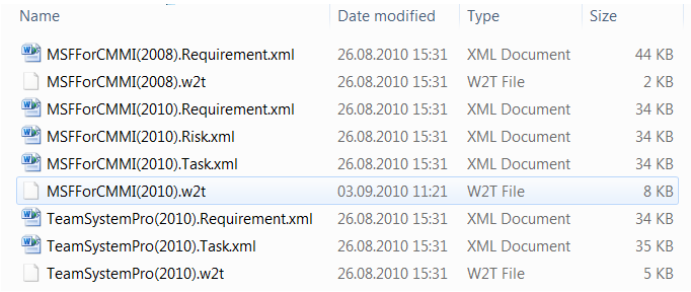


Figure 39 - UNC path



Name	Date modified	Type	Size
MSFForCMMI(2008).Requirement.xml	26.08.2010 15:31	XML Document	44 KB
MSFForCMMI(2008).w2t	26.08.2010 15:31	W2T File	2 KB
MSFForCMMI(2010).Requirement.xml	26.08.2010 15:31	XML Document	34 KB
MSFForCMMI(2010).Risk.xml	26.08.2010 15:31	XML Document	34 KB
MSFForCMMI(2010).Task.xml	26.08.2010 15:31	XML Document	34 KB
MSFForCMMI(2010).w2t	03.09.2010 11:21	W2T File	8 KB
TeamSystemPro(2010).Requirement.xml	26.08.2010 15:31	XML Document	34 KB
TeamSystemPro(2010).Task.xml	26.08.2010 15:31	XML Document	35 KB
TeamSystemPro(2010).w2t	26.08.2010 15:31	W2T File	5 KB

Figure 40 - The templates directory

The result:

Web locations

[Unfortunately, this isn't supported anymore – we're looking for customers that will fund this feature. Please contact: AIT.vertrieb@aitgmbh.de in case you want to finance its development.]

If you want to provide a worldwide available web location, you need a little bit of extra effort. The location folder is just the same as shown above, but you have to link to an additional mapping file as shown below. Link to this files full web path, like this:

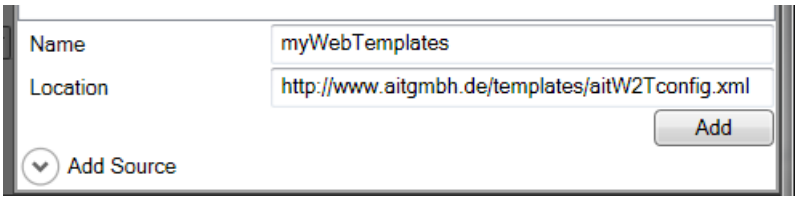


Abbildung 1 - Configuring a template location in the web

The referenced xml file contains a list of w2t files. Like the local or UNC location, these w2t files are the representation of the specific templates. The web xml file may look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfWebTemplatesConfigModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <WebTemplatesConfigModel Path="HTTP://www.aitgmbh.de/templates/MSFForCMMI(2010).w2t"/>
  <WebTemplatesConfigModel Path="HTTP://www.aitgmbh.de/templates/TeamSystemPro(2010).w2t"/>
</ArrayOfWebTemplatesConfigModel>
```

Figure 41 - An example of a configuration xml file

Team Project specific template

[This feature has not yet been implemented – it's a product idea only. We're looking for customers that will fund this feature. Please contact: AIT.vertrieb@aitgmbh.de in case you want to finance its development.]

Team project specific templates are set by checking in the template files into TFS source control as a project sub folder e.g. $/MyTeamProject/WordToTFSTemplates/myTemplate.w2t.

When a user connects with the team project via WordToTFS, the template checked into source control will be shown as available templates in WordToTFS only. All other templates will be hidden. This helps users using the right template and prevents them from having to configure a custom template location and choosing the template from the drop-down.

Project mapped template source.

Adding a location with subfolders will give you the option to set up the location as project mapped template source. This feature allows preselecting available templates based on what project the user is connected to. The expected folder structure relative to the location is “server name\team project collection name\project name”. A template placed in the subfolder “Server\DefaultCollection\Project A” will only be available if the user is connected to the project “Project A” within the team project collection “DefaultCollection” on the server “Server”.

Customizing the template configuration: w2t file

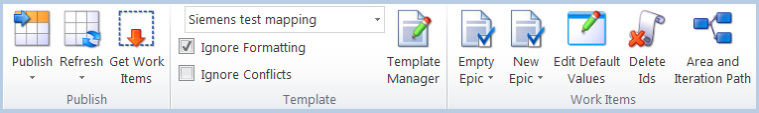
[In case you need further information on template customization or want to have a custom template implemented by us please contact: AIT.vertrieb@aitgmbh.de. We will bill any customizations based on a Remote Consulting Support budget.]

WordToTFS templates are fully customizable. Template configurations are XML files saved in files with a “w2t” extension. A template configuration may contain work item mapping configurations, header configurations and test report configurations.

A new configuration is defined using the MappingConfiguration XML element. A configuration defines behavior that applies to each work item and also allows enabling and disabling UI elements when this configuration is selected.

Table 1: Attributes of the MappingConfiguration element

Attribute	Description
ShowName	Name presented to the user.
UseStackRank (true|false)	Default: false Determines if there is a use for stack ranking work items – when set to True, all work items will get a stack rank that represents the order inside the currently published document. Publishing only selected work items may not work correctly, inserting a work item before other work items will change their stack rank an therefore create new revisions. You cannot simultaneously map the stack rank and set this property to true.
IgnoreFormatting (true|false)	Default: false Determines whether to compare only the contents (text) in HTML fields or to compare also formatting changes. Setting this to true will reduce the number of unnecessary published changes in HTML fields. If set to false, there might be ghost

	revisions (changes to TFS although no user change happened)
ConflictOverwrite (true|false)	Default: false Determines whether work items with revision conflicts are published anyways. Remote changes will be lost and overwritten with the values in the word document
EnableRefresh, EnablePublish, EnableGetWorkItems, EnableOverview, EnableHistoryComment, EnableEmtpy, EnableNew, EnableEditDefaultValues, EnableDeleteIds, EnableAreaIterationPath, EnableIgnoreFormattingSwitch, EnableConflictOverwriteSwitch (true|false)	Default: true Determines whether the button or checkbox is shown in the WordToTFS ribbon tab: 

Example:

```
<MappingConfiguration ShowName="TeamSystemPro(2010)"
                      RelatedSchema="MSFForCMMI.Schema.xsd"
                      DefaultMapping="false"
                      EnableRefresh="true"
                      UseStackRank="false"
                      IgnoreFormatting="true">
```

Creating a new work item mapping

A work item mapping relates to a single work item type (Requirement, Task, etc.) and defines what fields of that work item are mapped on what cells of a given work item table. It also defines how those fields are synchronized when the user refreshes or publishes a document. A work item mapping is created using the Mapping XML element within a MappingConfiguration XML element.

Table 2: Attributes of the Mapping XML element

Attribute	Description
WorkItemType	The TFS work item type (“Requirement”, “Bug”, etc.) this mapping applies to.
MappingWorkItemType	The name of the work item type in case you want to use a different name in your word document.
RelatedTemplate	The xml table style file. See below on how to customize these
AssignRegularExpression	A regular expression to find the *MappingWorkItemType.* The text you type in here must match text in a table cell in order for WordToTFS being able to identify the table as particular work item type.
AssignCellRow	The row of the cell where *AssignRegularExpression* is checked. The cell should contain the *AssignRegularExpression* as text in the template. Row count starts at 1.

AssignCellCol	The column of the cell where *AssignRegularExpression* is checked. The cell should contain the *AssignRegularExpression* as text in the template. Column count starts at 1.
ImageFile	An image to show in "new" button. If not customized, just type "standard.png"
WorkItemSubtypeField	This property offers the functionality to specify subtypes of the current Mapping
WorkItemSubtypeValue	The corresponding value for the *WorkItemSubtypeField*
HideElementInWord	Optional: Can be *true* or *false* and determines if this Mapping should be visible in the **New** and **Empty Work Item** buttons. (See: Chapter Support for any kind of work items)

Example:

```
<Mapping WorkItemType="Task"
         MappingWorkItemType="Task"
         RelatedTemplate="TeamSystemPro(2010).Task.xml"
         AssignRegularExpression="Task"
         AssignCellRow="1"
         AssignCellCol="1"
         ImageFile="standard.png">
        …
</ Mapping>
```

Defining field mappings

A field mapping defines what work item field is mapped onto which cell of a template and when to refresh or publish the value of this cell. Field mappings are created using the *Field* XML element within the *Mapping* XML element. The fields **System.ID** and **System.Rev must be mapped** in order for WordToTFS to work correctly!

Attribute	Description
Name	*ReferenceName* of the field in your process template, such as 'System.Id'
FieldValueType	The type of this field. This setting mostly depends on the data type of this field in your process template. For details see below.
Direction	The direction in which to synchronize this field. For possible values and their meaning, see table below.
MappingTableRow	The row of the word table cell this field is mapped to. Row count starts at 1.
MappingTableCol	The column of the word table cell this field is mapped to. Column count starts at 1.
TestCaseStepDelimiter	A string that separates test step title and test step expected result when mapping test steps of test case work items. Default value is '->'.
ShapeOnlyWorkaroundMode (ShowAsError |AddSpace)	*ShowAsError* will show an error when trying to publish a field with only an inline shape (picture) and no text. Without text word will not export shapes. *AddSpace* will automatically add a space after

	the inline shape instead of showing an error.

Example:
This configuration maps the id of a work item to the first column in the second row of a table. The id is only read from the server, but never written to.

```
<Field Name="System.Id"
       FieldValueType="PlainText"
       Direction="TfsToOther"
       MappingTableRow="3"
       MappingTableCol="1" />
```

Direction Attribute

The exact behavior of when field values are published or refreshed is given by the following table:

	Get	Refresh	Publish
GetOnly	Value is refreshed	-	Value is refreshed once for work items that have been created in Word.
TfsToOther	Value is refreshed	Value is refreshed	Value is refreshed
OtherToTfs	Value is refreshed	Value is refreshed	Value is **published**, then refreshed
PublishOnly	-	-	Value is **published**
SetInNewTfsWorkItem	Value is refreshed	Value is refreshed	Value is **published** once for work items that have been created in word. Value is refreshed.

FieldValueType Attribute

PlainText

The field value is interpreted as plaint text both in Word and in the team foundation server. Use this value for all fields that are configured as simple data types like *Integer*, *String*, etc. in your process template.

HTML

The field value is interpreted as HTML both in Word and in the team foundation server. Use this value for all fields that are configured as *HTML* data type in your process template. If you do not need the HTML formatting options and have encountered problems you might want to use *PlainText* instead.

DropDownList

The field value is interpreted as Dropdown-List in Word. Use this value for all fields that are configured with *AllowedValues*, either explicitly or implicitly (e.g. '*System.AssignedTo'*) in your process template.

BasedOnFieldType

The field value is interpreted as numbered list of test case titles and test case expected results,

separated by the *TestCaseStepDelimiter.* Use this value for all fields that are configured as *TestStepsControl* data type in your process template (e.g. '*Microsoft.VSTS.TCM.Steps*'). Please note that formatting of test steps is not supported in WordToTFS and will be lost during import or publish!

Example configuration:

```
<Field Name="Microsoft.VSTS.TCM.Steps"
       FieldValueType="BasedOnFieldType"
       Direction="OtherToTfs"
       MappingTableRow="3"
       MappingTableCol="1"
       TestCaseStepDelimiter="->" />
```

Test Case	2	Sven Hubert	Design	
First Test			311	3

1. Do that
2. Do this -> Check that
3. Do more
4. Close all

Figure 42 – Work item with mapped test case test steps in Word

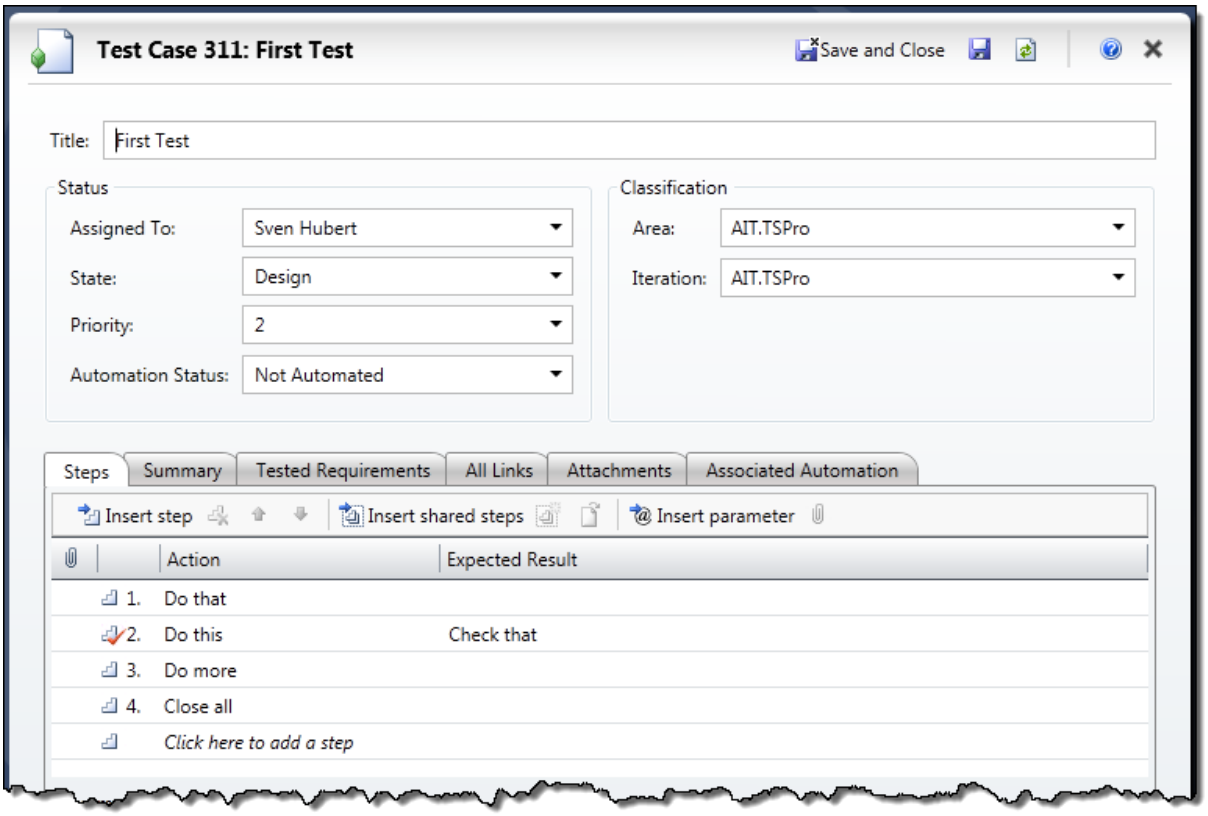


Figure 43 - Work item with test steps in MTM

Default value

If you want to provide a default value to a cell/field, you have to add the sub node “DefaultValue”. On creating a new work item at word, these values are used to prefill the table, and you also be able to edit the default values through the “Edit default Values” button.

Attribute	Description
Show Name	The label used in default values pane
>Value<	The default value to use

Example:

```
<DefaultValue ShowName="Title">Task Title</DefaultValue>
```

OLE Support

By default, OLE objects are converted into images which are attached to the work item. This behavior can be changed by setting the attribute ‘HandleAsDocument’ to true. This causes the complete field contents to be attached as micro Word document to the work item. The work item field will contain a view in HTML which should not be changed by the user in TFS as it will be overwritten during the next publish in Word. The micro Word document contains all original OLE objects and is used during import and refresh in Word.

The attribute ‘HandleAsDocumentMode’ can be set to ‘OleOnDemand’ which reduces the behavior of attaching a micro Word document to the cases where OLE objects are actually contained in the work item table.

Example:

```
<Field Name="System.Description"
       FieldValueType="HTML"
       Direction="OtherToTfs"
       MappingTableRow="3"
       MappingTableCol="1"
       HandleAsDocument="true"
       HandleAsDocumentMode="OleOnDemand">
         ...
</Field>
```

Converters/Converter

A converter can help you to show the values in the right way.

Attribute	Description
Field Name	TFS field name to convert
Text	TFS Value
MappingText	Word Value

Example:

```
<Converters>
  <Converter FieldName="Microsoft.VSTS.Common.Severity">
    < Values>
      <Value Text="1 - Critical" MappingText="1" />
      <Value Text="2 - High" MappingText="2" />
```

```
      <Value Text="3 - Medium" MappingText="3" />
      <Value Text="4 - Low" MappingText="4" />
      <Value Text="4 - Low" MappingText="" />
    </Values>
  </Converter>
</Converters>
```

Defining link mappings

Using the links element and the link element, you can set up table cells to contain formatted work item links.

Attribute	Description
Type	Name of the relation. For example "**child"** if you want to edit children in this field, "**parent"** if you want to edit the parent. You can use any registered link type.
Direction	Synchronization direction. See table
MappingTableRow	Row of the field within the template.
MappingTableCol	Column of the field within the template.
Overwrite	Sets whether the links in the mapped cell overwrite existing links. **"False"** means, links can only be added. "**True"** means, links deleted in Word are also deleted on your TFS. This attribute is only meaningful when combined with directions "OtherToTfs" or "PublishOnly".
LinkSeparator	Character sequence used to separate multiple entries. Use \t or \r for tab or return respectively
LinkFormat	A composite format string used to format output for a single link. You can access any field of the referenced work item by "**{FieldName}"**
AutoLinkWorkItemType	Setting this to a valid work item type will automatically add a link to the last work item of the given type that appears before the current work item in the Word document.
AutoLinkWorkItemSubtypeField	Setting this to a valid field reference name will automatically add a link to the last work item of type "AutoLinkWorkItemType" that appears before the current work item in the Word document and whose "AutoLinkWorkItemSubtypeField" has the value "AutoLinkWorkItemSubtypeValue".
AutoLinkWorkItemSubtypeValue	Setting this to a valid value will automatically add a link to the last work item of type "AutoLinkWorkItemType" that appears before the current work item in the Word document and whose "AutoLinkWorkItemSubtypeField" has the value "AutoLinkWorkItemSubtypeValue".
AutoLinkSuppressWarnings	Setting this to "False" will suppress warnings if automatic linking was defined but no appropriate work item was found in the document.

	Get	Refresh	Publish

GetOnly	Work item links are refreshed	-	Work item links are refreshed for new work items
TfsToOther	Work item links are refreshed	Work item links are refreshed	Work item links are refreshed
OtherToTfs	Work item links are refreshed	Work item links are refreshed	Work item links are **published**, then refreshed
PublishOnly	Work item links are refreshed	-	Work item links are **published**
SetInNewTfsWorkItem	Work item links are refreshed	Work item links are refreshed	Work item links are **published** for new work items

Example 1

```
<Links>
  <Link Type="Child"
        Direction="OtherToTfs"
        MappingTableRow="2"
        MappingTableCol="2"
        Overwrite="false"
        LinkFormat="{System.Id} - {System.Title}" />
</Links>
```

This example configures the second cell in the second row of the table to display all links to all children in the format “1929 – Title”. If the work item has multiple children, each entry is display on a new line. Changes made in word will be published, but only if they add children. Deleting children will not be possible.

Example 2

```
<Links>
  <Link Type="Parent"
        Direction="OtherToTfs"
        MappingTableRow="2"
        MappingTableCol="2"
        Overwrite="true"
        AutoLinkWorkItemType="Requirement"
        AutoLinkWorkItemSubtypeField="Microsoft.VSTS.CMMI.RequirementType"
        AutoLinkWorkItemSubtypeValue="Business Objective"
        LinkFormat="{System.Id} - {System.Title}" />
</Links>
```

This example configures the second cell in the second row of the table to display a link to the parent work item. When publishing, WordToTFS searches from the current work item to the top of the document for the first Requirement of type Business Objective and links to that work item.

Be aware of the following restrictions

- The link format must always contain the field reference “{System.Id}”.
- The link separator must not be contained in either the link format string or the referenced fields. For example using a comma as link separator will not work if one of the referenced fields contains a comma as well.
- When referencing multiple fields like System.Id and System.Title in the example, make sure that the character sequence between the referenced fields does not appear in either of the

separated field values. The above example would not work if the System.Title contains the sequence " – ".

Nodes <Transitions>, <Transition>

This elements can be used to configure an automatic state transition during publish.

Attribute for <Transitions> and <Transition>

Attribute	Description
FieldName	Reference name of the field that should be altered.
From	Identifier of the current state
To	Identifier of the next state

In some cases, fields are read-only for a work item in its current state. If by changing these fields in Word you mean to imply that the state of the work item also changes, you can use automatic state transitions to define what state to transition to:

Example:

```
<Transitions FieldName="System.State">
  <Transition From="Released" To="Draft" />
  <Transition From="Approved" To="Draft" />
</Transitions>
```

This example shows how to reset the state field of a work item back to "Draft" if the user makes changes to the mapped work item. The referenced state field (here "System.State") does not have to be configured using the <Field> element, but if it is then:

1. The value changed in Word, overrides the transition rule (the user value wins)
2. The Direction of the field must be set to "OtherToTfs"

Defining Headers

Headers are defined within <Header></Header> tags in the MappingConfiguration/Headers section of the configuration file:

```
<MappingConfiguration>
  <Headers>
    <Header Identifier … >
    </Header>
  </Headers>
  <Mappings>
  </Mappings>
</MappingConfiguration>
```

The <Header>-Element has the following Attributes:

Attribute	Description
Identifier	Name of the header.
Row	Row of the cell in which to look for the identifier

Column	Column of the cell in which to look for the identifier
Level	The hierarchy level of the header. Field definitions of a header are overwritten by headers of higher level (within their range) or by work items if mapped. A Header will affect work items until another header of the same or lower level.
RelatedTemplate	As in <Mapping>
ImageFile	As in <Mapping>

Within the <Header>-Tag you can define Field mappings as for normal work item mappings. You can also define converters, but you cannot define state transitions or links. The only supported directions for fields in headers are *SetInNewTfsWorkItem* and *WordToTfs*.

If the same field is mapped by multiple headers within the same range, only the last header definition applies. If the same field is mapped by a header and an affected work item, publishing result depends on the field direction. If it is set to *WordToTfs*, the value is always read from the header and the header configuration applies. **This will overwrite changes made to individual affected work items**. If an affected field is changed from outside word, the usual conflict checking mechanism applies. If the header value conflicts with a newer value of an affected work item, you have to refresh the work item first, but the changes will then be overwritten with the header field.

If the direction is set to *SetInNewTfsWorkItem*, the value is read from the header and the configuration of the header applies for all affected work items where no value is set. If any value is set in the work item, the work item configuration applies and the value of the work item is published. **In any case, all subsequent operations will use the work item field configuration**.

3: Example header configuration

```
<Header Identifier="DocumentHeader"
        Level="1" Row="1" Column="1"
        RelatedTemplate="TeamSystemPro(2010).DocumentHeader.xml"
        ImageFile="standard.png">
  <Fields>
    <Field Name="Microsoft.VSTS.Common.Priority"
           Direction="OtherToTfs"
           FieldValueType="PlainText"
           MappingTableRow="2"
           MappingTableCol="2" />
    <Field Name="System.AssignedTo"
           Direction="OtherToTfs"
           FieldValueType="PlainText"
           MappingTableRow="3"
           MappingTableCol="2" />
  </Fields>
  <Converters>
    <Converter FieldName="Microsoft.VSTS.Common.Priority">
      <Values>
        <Value Text="1" MappingText="Critical" />
        <Value Text="2" MappingText="High" />
        <Value Text="3" MappingText="Medium" />
        <Value Text="4" MappingText="Low" />
      </Values>
    </Converter>
```

```
  </Converters>
</Header>
```

Customizing the template layout: .xml files

The Layout and design is provided through table xml files. To generate these files, just create a blank word file with a table, and save it as “Word Xml File”. The simplest work item table is a simple table with some rows and cols. But you also can use background color, merge cells (only horizontally) , invisible borders, font size, type and color and so on. Save them in the same folder as your other template stuff, the w2t file and the optional favicon, and generate the entry for the file at the w2t mapping definition. Ensure that the filename is written right at the *RelatedTemplate* – Tag at the Mapping – Node.

Additional you have to make sure you have the right amount of cells. As example, your word file contains a table with 5 columns and 2 rows as shown below, you can´t map a TFS field to col 1 with row 3.

row 1, col 1	row 1, col 2				row 1, col 3
	row 2, col 2	row 2, col 3			row 2, col 6
Nonexiting cell					

Configuring the test reporting extension

WordToTfs supports generating test specification reports and test result reports. The specification report is intended to generate an overview of how the tests were conducted while the result report gives an overview of the latest test outcomes. The general structure of the generated document is fixed, but the individual components can be customized independently for both report types. Typically more verbose information like test setup is shown in the specification report and the result report only lists the test case title and id for a test case along with its test outcomes. It is of course possible to use the same component configuration in both report types.

The basic configuration looks like this

```
<MappingConfiguration>
   <TestConfiguration>
      <TestSpecificationConfiguration … />
      <TestResultConfiguration … />
      <Templates>
         …
      </Templates>
   </TestConfiguration>
</MappingConfiguration>
```

`TestResultConfiguration` and `TestSpecificationConfiguration` define what templates are used for the various components that make up a report. The templates themselves are defined within the `Templates` – Section

General test configuration

```
<TestConfiguration SetHyperlinkBase="true"/>
```

Attribute	Explanation
SetHyperlinkBase (true|false)	When enabled, the document hyperlink base is set to the current document location. Turn this on if you use attachment downloading but your hyperlinks do not work. Hyperlinks will stop working for word installations without WordToTfs plugin after the document is moved to a different location if this option is turned on.

Test specification report configuration

```
<TestSpecificationConfiguration
    Available="true"
    TestPlanTemplate="TestPlan"
    TestSuiteTemplate="TestSuite"
    TestCaseElementTemplate="SpecificationReportTestCaseRow" />
```

Attribute	Explanation
Available (true|false)	Shows (true) or hides (false) the button in the word ribbon, thereby allowing or preventing users from generating test specification reports.
TestPlanTemplate	Name of a template configured in the templates section that is used to generate test plan information. If this attribute is missing no test plan information is shown in the report.
TestSuiteTemplate	Name of a template configured in the templates section that is used to generate test suite information. If this attribute is missing no test suite information is shown in the report.
TestCaseElementTemplate	Name of a template configured in the templates section that is used to generate test case information. If this attribute is missing no test case information is shown in the report.

Test result report configuration

```
<TestResultConfiguration
    Available="true"
    TestPlanTemplate="TestPlan"
    TestSuiteTemplate="TestSuite"
    TestCaseElementTemplate="ResultReportTestCaseRow"
    TestResultElementTemplate="ResultReportTestResultRow"
    TestConfigurationElementTemplate="ResultReportTestConfigurationRow" />
```

Attribute	Explanation
Available (true|false)	Shows (true) or hides (false) the button in the word ribbon, thereby allowing or preventing users from generating test specification reports.
TestPlanTemplate	Name of a template configured in the templates section that is used to generate test plan information. If this attribute is missing no test plan information is shown in the report.
TestSuiteTemplate	Name of a template configured in the templates section that is used to generate test suite information. If this attribute is missing no test suite information is shown in the report.
TestCaseElementTemplate	Name of a template configured in the templates section that is used to generate test case information. If this attribute is

	missing no test case information is shown in the report.
TestResultElementTemplate	Name of a template configured in the templates section that is used to generate test result information. If this attribute is missing no test result information is shown in the report.
TestConfigurationElementTemplate	Name of a template configured in the templates section that is used to generate test configuration information. If this attribute is missing no test configuration information is shown in the report.

Templates

A template defines a view on an object of the tfs object model. It consists of the actual view that is an external word document (typically a table) and information on how to bind the view to the object properties. The external document is inserted into the generated report whenever information of the object the view is defined for should be reported. The inserted document is then searched for bookmarks and each bookmark is replaced with the value of a property as configured in the template.

```
<Template Name="TestPlan" HeaderTemplate=" " FileName="TestPlan.xml">
  <PreOperations>
    <Operation Type="DeleteCharacterLeft"/>
  </PreOperations>
  <PostOperations>
    <Operation Type="MoveCursorToLeft"/>
  </PostOperations>
  <Replacements>
    <Replacement Bookmark="Name" Property="Name"/>
    <Replacement Bookmark="AreaPath" Property="AreaPath"/>
    <Replacement Bookmark="Iteration" Property="Iteration"/>
    <Replacement Bookmark="Description" Property="Description" ValueType="HTML"/>
    <Replacement Bookmark="StartDateDate" Property="StartDate.Date"/>
    <Replacement Bookmark="EndDateDate" Property="EndDate.Date"/>
    <Replacement Bookmark="BuildDirectory" Property="BuildDirectory"/>
    <Replacement Bookmark="RootSuiteTitle" Property="RootSuite.Title"/>
    <Replacement Bookmark="OwnerDisplayName" Property="Owner.DisplayName"/>
  </Replacements>
</Template>
```

This example configuration instructs the generation algorithm to insert the file “TestPlan.xml” into the document when the document structure requires showing test plan information. It then tries to replace the bookmark “Name” with the name of the test plan, the bookmark “AreaPath” with the area path of the test plan etc.

Attribute/Node	Explanation
Template	Node for the definition of one node.
Name	User defined unique name that is used to identify the template.
HeaderTemplate	Name of template to use as header. The name is used only in such case, if the actual template is used to generate a block of the elements of the same type. This feature is usable to generate a table with header.
FileName	The name of the file used as Word template inserted on the required location in generated report. (The structure of generated report is described in chapter “Structure of generated report”.)
Replacement	The definition for a single replacement. The bookmark in inserted template will be replaced by value of evaluated

	property.
Bookmark	Name of the bookmark to replace.
Property	The property of the bound object that replaces the bookmark. The property can be nested – 'RootSuite.Name'. If the path contains a string indexer, the key must be enclosed in double quotation marks: 'CustomFields["AIT.Description"].Value'
ValueType	Type of the value of the evaluated property. Possible values are "PlainText" and "HTML". If this attribute is omitted, the value "PlainText" is used.
PreOperations **PostOperations**	The definition of an array of operation that should be performed prior to insertion of the template, after inserting the template to target document. The Operation is performed in target document.
Operation	Type of the operation defines the operation to execute. Possible types: • InsertParagraph • MoveCursortToStart • MoveCursortToEnd • DeleteCharacterLeft • DeleteCharacterRight • MoveCursorToLeft • MoveCursorToRight

Invoke methods from within the template

In addition to bind Bookmark to properties, it is also possible to invoke a method on the property before the value is inserted into the document. If you want a custom date format, you can invoke the ToString-Method and pass the formatting string in parantheses. Note that you have to use single quotation marks for the property if you want to pass a string argument to the method.

```
<Template … >
  <Replacements>
    <Replacement Bookmark="Date" Property='Date.ToString("dd-mmm-YYYY")'/>
  </Replacements>
</Template>
```

Object used to resolve property value

We have defined that 4 templates for Test specification report can be defined and 8 templates for Test result report. For every template is dedicated one object from Microsoft object model in Team Foundation Server. Therefore the user must look at the corresponding documentation, if he wants to define the properties that need to be evaluated in replacement phase of creating of report.

Table 4: Base objects used in replacement phase of generation of report

Template	Interface from object model
Test plan	ITestPlan (Link to interface on MSDN)
Test suite	ITestSuiteBase (Link to interface on MSDN)
Test case	ITestCase (Link to interface on MSDN)
Test result	ITestCaseResult (Link to interface on MSDN)
Test configuration	ITestConfiguration (Link to interface on MSDN)

Examples of properties in ITestCase:

- WorkItem.Id – identification of the test case

- WorkItem.Rev – revision of the test case
- WorkItem.Title – title of the test case
- Actions – actions / steps in test case – formatted output of array is described in chapter "**Linked templates**".

Object model extensions

In addition to the properties found in the MSDN documentation, some objects are extended by properties we think make the configuration easier or are outright lacking:

Interface	Extended with
ITestStep	StepNumber(int) – The number of the step as shown in TFS test case work item. Shared steps are inlined, therefore this value may be different for the same shared step for different test cases.
ITestStepResult	StepNumber(int) – The number of the test step this for this result as shown in TFS test case work item. Shared steps are inlined, therefore this value may be different for the same shared step in different test cases. Title(string) – The title of the tested test step. Parameters are replaced with the values of the tested iteration. ExpectedResult(string) – The expected result of the tested test step. Parameters are replaced with the values of the tested iteration. TestStep(TestStep-Object) – The tested test step.
ISharedStep	Not available. Shared steps are automatically inlined.
ISharedStepResult	Not available. Shared step results are automatically inlined.
ITestAttachment	LocalPath (string) – Path to the local file if the attachment was downloaded. Only available in combination with the AttachmentLink-element LocalFilename (string) – Filename of the local file if the attachment was downloaded. Only available in combination with the AttachmentLink-element

Hyperlink configuration

In this test reporting extension is implemented support for hyperlinks:

- Common hyperlink
- Hyperlink to viewer of build on Web Access portal of project.
- Hyperlink to editor of work item on Web Access portal of project
- Hyperlink to viewer of work item on Web Access portal of project
- Hyperlink to viewer of work item in specific revision on Web Access portal of project
- Hyperlink to view of test attachments

Therefore is an extension of replacement configuration defined.

Common hyperlink

```
<Replacement Bookmark="AttachmentName" Property="Name">
    <UriLink Uri="Uri"/>
```

```
</Replacement>
```

As Uri must be defined the full Uri. For example an Uri of attachment.

Hyperlink for viewer of build

```
<Replacement Bookmark="BuildNumber" Property="BuildNumber">
    <BuildViewerLink BuildNumber="BuildNumber"/>
</Replacement>
```

As BuildNumber must be defined the build number.

Hyperlink for editor of work item

```
<Replacement Bookmark="IdEdit" Property="WorkItem.Id">
    <WorkItemEditorLink Uri="WorkItem.Uri"/>
</Replacement>
```

As Uri must be defined the team foundation server artifact uri.

Hyperlink to viewer of work item

```
<Replacement Bookmark="IdView" Property="WorkItem.Id">
    <WorkItemViewerLink Id="WorkItem.Id"/>
</Replacement>
```

As Id must be defined the id of work item.

Hyperlink to viewer of work item in specific revision

```
<Replacement Bookmark="IdRev" Property="WorkItem.Id">
      <WorkItemViewerLink Id="WorkItem.Id" Revision="WorkItem.Rev"/>
</Replacement>
```

As Id must be defined the id of work item and as Revision the revision of work item.

Hyperlink to viewer of test attachments

The occurrence of an attachment link will trigger downloading the attachment to a subfolder of the document for the modes “DownloadAndLinkToLocalFile” and “DownloadOnly”. The properties “LocalPath” and “LocalFilename” are only available using these modes. Using the mode “DownloadOnly” in combination with a property other than “LocalPath” or “LocalFilname” is not recommended as the attachments will be downloaded but there is usually no way to tell which a attachments belongs to which occurrence in the document.

```
<Replacement Bookmark=”PropertyBookmark” Property=”LocalPath”>
      <AttachmentLink Mode="DownloadAndLinkToLocalFile"/>
</Replacement>
```

Linked templates

As previously mentioned, in the configuration can be defined the linked template. This means that bookmark is not replaced with simple text (evaluated property), but with other template.

The configuration telling that the linked template is to use, should be defined as follows:

```
<Replacement Bookmark="ActionList"
             Property="Actions"
             LinkedTemplate="ActionSimpleList"/>
```

And the definition of template can be defined as follows:

```
<Template Name="ActionSimpleList" FileName="…">
    <Replacements>
        <Replacement Bookmark="Title" Property="Title.ToPlainText"/>
        <Replacement Bookmark="ExpectedResult"
Property="ExpectedResult.ToPlainText"/>
    </Replacements>
</Template>
```

FAQ

Q Installation fails with "The following Microsoft Office Solution cannot be loaded because a compatible version of the .NET Framework is not installed.

A Make sure you have both .NET 3.5 SP1 and .NET 4.0 installed

Q Why do I get "Revision conflicts" when trying to publish although I have refreshed to the most recent version and I am sure nothing has changed on the TFS?

A You may have forgotten to map the System.Rev field which is necessary for this check to work correctly. Also you may have selected a template that uses a "PublishOnly" direction which means your word work item will never be refreshed to the latest version.

Q Why is text formatting sometimes not published to the TFS when using HTML fields?

A Word exports some formatting only when it considers the item finished. This is not automatically the case for the last formatting item within a cell. In this case, you have to make sure that the last symbol in an end-of-paragraph marker (by pressing Enter)

Q Import / Export of items does not work for a specific work item but it does for others

A Make sure, the table layout of the affected work item does not contain vertically merged cells. Cell addressing by row and column does not work with vertically merged cells.

Q Template is not shown in Word

A In case a custom template is not shown in the list of available Template, verify that the template location is added using the template manager and that the w2t is correct. In case of syntax or logical errors, the template will not be shown as available template in Word.

Copyright

The tool AIT - WordToTFS User Guide is provided by AIT Applied Information Technologies GmbH & Co. KG, Leitzstr. 45, 70469 Stuttgart, Germany.

The content of this document is the intellectual property of the AIT GmbH & Co. KG. © 2010 AIT GmbH & Co. KG

AIT Applied Information Technologies GmbH & Co. KG
AIT TeamSystemPro Team

Email	info@aitgmbh.de
Internet	www.aitgmbh.de/teamsystempro
Phone	+49 711 49066 430
Fax	+49 711 49066 440

Postal address:	General Partner:	CEO: Lars Roith
Leitzstr. 45	AIT Verwaltungs GmbH	
70469 Stuttgart	Amtsgericht Stuttgart	IBAN: DE80 61191310 0664310001
Deutschland	HRB 734136	SWIFT: GENODES1VBP

This document is protected by German copyright laws and may only be reproduces, modified or extended with the written consent of the authors. All Rights reserved.